Diplomski studij

Informacijska i komunikacijska tehnologija

Telekomunikacije i informatika

Računarstvo

Programsko inženjerstvo i informacijski sustavi

Računalno inženjerstvo
Računarska znanost

Ak. g. 2016./2017.

Raspodijeljena obrada velikih skupova podataka

1. Laboratorijska vježba

Zadatak 1: Upoznavanje sa Hadoop grozdom

Cilj ovog zadatka jest upoznavanje sa Apache Hadoop grozdom kojeg ćete koristiti na laboratorijskim vježbama a koji je postavljen u obliku Cloudera Apache Hadoop distribucije.

Priprema platforme

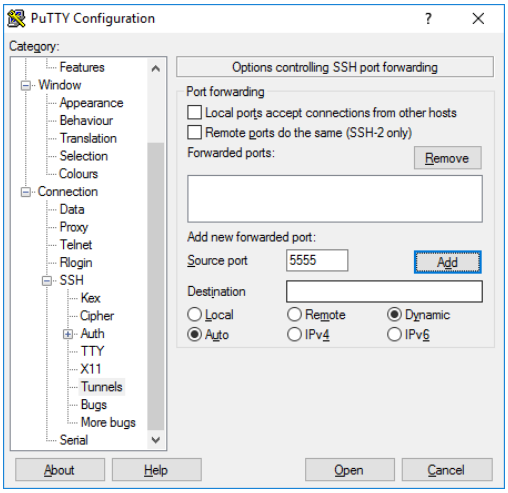
Za početak je potrebno konfigurirati radnu platformu, za što će vam trebati sljedeći softver (pretpostavka je OS Windows, za ostale operacijske sustave koristite se softverom analogne funkcionalnosti):

- **Java SE Development Kit 8** (Java platforma)
- **Netbeans IDE** (Java razvojno sučelje)
- **Putty** (znakovno sučelje prema udaljenim računalima)
- **WinSCP** (prijenos podataka prema udaljenim računalima)
- **Internet pretraživač** (preporučujemo Firefox)

Nastavak uputa pretpostavit će korištenje preporučenog softvera. Pratite pažljivo sljedeće korake:

1) Otvorite **putty** i stvorite novu konekciju sa sljedećim parametrima (nemojte se još spajati!):
   Hostname: rovkp@iotat.tel.fer.hr
   Port: 50051

2) Otiđite na *Connection -> SSH -> Tunnels* i složite *port forwarding* na način kako prikazuje slika:



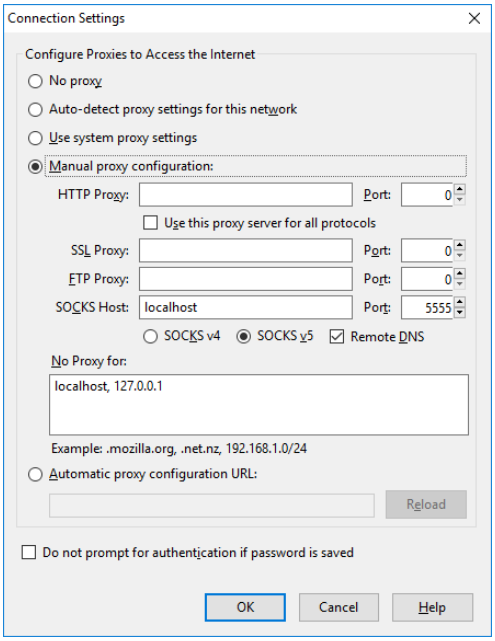
3) Snimite parametre konekcije i spojite se kao korisnik *rovkp* (standardna lozinka).

4) Unesite naredbu *ifconfig* i pogledajte IP adresu računala (za nastavak ćemo pretpostaviti da je IP adresa koju vidite 10.9.4.51).

5) Otvorite internetski pretraživač (preporučujemo Firefox, za druge pretraživače potražite postavke analogne onima prikazanima u narednim koracima)

6) Otiđite na *Tools -> Advanced -> Network -> Settings -> Manual Proxy Configuration*

7) Zapamtite eventualne trenutne postavke kako bi ih mogli vratiti u originalno stanje nakon laboratorijskih vježbi. Unesite postavke na način kako prikazuje slika i stisnite *OK*.



8) U adresni prostor pretraživača unesite
   10.19.4.51:7180

9) Ukoliko ste pažljivo pratili navedene korake, trebali biste vidjeti naslovnu stranicu *cloudera manager*-a. U slučaju da istu ne vidite vratite se na prethodne korake i potražite eventualne pogreške u proceduri.

10) Prijavite se kao korisnik *rovkp* sa standardnom lozinkom.

Ovime ste uspješno konfigurirali platformu za izvršavanje zadatka.

NAPOMENA: odgovorno koristite web sučelje i ne unosite nikakve izmjene koje mogu utjecati na rad grozda!

Zadatak

Proučite funkcionalnosti web sučelja, ponuđene grafove i opcije.

Uz pomoć web sučelja odgovorite na sljedeća pitanja:

1) Od koliko računala se sastoji ovaj Hadoop grozd?

2) Koja je IP adresa računala koje ima ulogu imenskog čvora (*Namenode)*?

3) Koliko ima podatkovnih čvorova (*Datanodes)*? Jesu li svi u funkciji?

4) Koji je ukupni spremišni kapacitet grozda?

5) Koja je postavljena veličina HDFS bloka?

Zadatak 2: Rad sa datotekama u Hadoop grozdu

(*NAPOMENA: zadatak se temelji se na zadacima 2. i 3. iz prve domaće zadaće. Možete se koristiti rješenjima domaće zadaće kao predložak za rješenje ovog zadatka)*

Priprema platforme

Uz pomoć programa *putty* spojite se na udaljeno računalo *cloudera* grozda koje će vam služiti kao rubni čvor (IP adresu računala će vam dati asistent, korisničko ime je *rovkp* a lozinka standardna).

Na lokalnom datotečnom sustavu udaljenog računala stvorite mapu *iprezime* (npr. ako se zovete Iva Horvat stvorite mapu čija će apsolutna staza biti */home/rovkp/ihorvat*).

Stvorite mapu istog imena na HDFS-u u podmapi */user/rovkp* (znači Iva Horvat će na HDFS-u stvoriti mapu */user/rovkp/ihorvat*).

Ove dvije mape biti će Vaše radne mape udaljenog računala/Hadoop grozda u zadacima 2 i 3!

Otvorite WinSCP (ili neku drugu aplikaciju za prebacivanje datoteka na udaljeno računalo). Spojite se na radnu mapu udaljenog računala.

Provjerite možete li uspješno prebaciti datoteku sa lokalnog datotečnog sustava radnog računala na lokalni datotečni sustav udaljenog računala. Također, provjerite da li možete prenijeti datoteku sa lokalnog datotečnog sustava udaljenog računala na HDFS i obrnuto.

Pokrenite NetBeans i stvorite novi *Maven* projekt.

Ovime ste obavili sve pripremne korake potrebne za izradu vježbe.

Zadatak

Cilj zadatka je napisati Javin program za spajanje kolekcije tekstualnih datoteka u jedinstvenu tekstualnu datoteku. Knjige su pohranjene u obliku tekstualnih datoteka na lokalnom datotečnom sustavu. Ciljna datoteka mora se pohraniti na HDFS-u.

Uspješnim rješenjem ovog zadatka steći ćete sljedeća znanja:

- **razvoj i izvršavanje programa u razvojnom sučelju NetBeans predviđenih za izvođenje na Hadoop grozdu**
- **rad sa tokovima podataka lokalnog i HDFS datotečnog sustava**

Detaljni opis zadatka

Uz pomoć naredbe `wget` dohvatite datoteku `gutenberg.tar.gz` (veličine 151 MB) sa sljedeće poveznice:

http://svn.tel.fer.hr/gutenberg.tar.gz

Raspakirajte sadržaj datoteke u lokalni datotečni sustav Hadoop grozda. Obrišite datoteku *gutenberg.tar.gz*.

Pogledajte strukturu raspakirane datoteke – ukoliko ste sve uspješno izveli vidjet ćete novu mapu *gutenberg* i unutar nje niz podmapa (naziva `etext00`, `etext01` itd.) koje sadrže niz tekstualnih datoteka (knjiga na engleskom jeziku).

Napišite Javin program koji mora iterativno proći kroz ovu strukturu i podatke iz svih tekstualnih datoteka pohraniti u novu datoteku *`gutenberg_books.txt`* koja će se nalaziti na HDFS-u Sve izvorne datoteke moraju se čitati i prenositi redak po redak.

(ukoliko imate problema sa programskim pretraživanjem direktorija, pogledajte metodu listStatuses klase FileSystem koja vraća polje objekata tipa FileStatus a koje možete pretvoriti u stazu uz pomoć metode getPath())

*NAPOMENA: Ciljna datoteka mora se stvoriti programski. Rješenja gdje se ciljna datoteka stvori lokalno a onda uz pomoć komandne linije prenese na HDFS **neće se priznavati**.*

Nakon stvaranja ciljne datoteke program mora na zaslon ispisati sljedeće informacije:

- koliko je ukupno datoteka pročitano
- koliko je ukupno redaka pročitano
- koliko vremena (u sekundama) je bilo potrebno za izvođenje programa

Dodatno, odgovorite na sljedeća pitanja:

- koja je veličina ciljne datoteke?
- u koliko je ukupno blokova pohranjena na HDFS-u (uključujući i replikaciju)?

NA KRAJU LABORATORIJSKE VJEŽBE OBRIŠITE MAPU gutenberg SA SVIM PODMAPAMA I DATOTEKU gutenberg_books.txt NA HDFS-u (TAKOĐER I DATOTKU gutenberg.tar.gz AKO STE JE ZABORAVILI OBRISATI TOKOM PROVEDBE ZADATKA)!

* * *

Zadatak 3: Serijalizacija objekata u Hadoop grozdu

Prije rješavanja zadatka pročitajte uvodni dio koji će objasniti osnovne koncepte serijalizacije u Hadoop okruženju te način korištenja datoteke predviđene za sekvencijalnu pohranu parova ključ-vrijednost.

UVOD

Serijalizacija je proces pretvaranja objekata u niz bajtova pogodan za prijenos preko mreže ili trajnu pohranu. Obrnuti proces zovemo *deserijalizacija*.

Hadoop koristi svoj vlastiti format za serijalizaciju zvan *Writables*. Objekti koje želimo serijalizirati unutar *Hadoop* infrastrukture moraju implementirati sučelje *Writable* (kako bi se mogli serijalizirati i deserijalizirati), a često i *Comparable* (kako bi se mogli koristiti kao ključevi). Hadoop također nudi i svoje inačice primitivnih Javinih tipova (*IntWritable* umjesto *int*, *FloatWritable* umjesto *Float*, *Text* umjesto *String* i sl.) koji implementiraju *Writable* sučelje.

Pohrana podataka u obliku ključ-vrijednost je sveprisutna u *Hadoop* infrastrukturi, a za te potrebe *Hadoop* između ostalog definira i Javinu klasu *SequenceFile* koja omogućuje sekvencijalnu pohranu parova ključ-vrijednost, gdje su i ključ i vrijednost *Writable* objekti. Ovakva datoteka podesna je za naknadne obrade unutar *Hadoop* infrastrukture (npr. za *MapReduce* obradu). *SequnceFile* možemo stvoriti uz pomoć klase *SequenceFile.Writer* na sljedeći način:

```
SequenceFile.Writer writer = SequenceFile.createWriter(conf,
      SequenceFile.Writer.file(outputPath),
      SequenceFile.Writer.keyClass(key.getClass()),
      SequenceFile.Writer.valueClass(value.getClass());
```

gdje je *conf* objektna reprezentacija *Hadoop* konfiguracije, *path* objektna reprezentacija datoteke, a *key* i *value* objekti koji predstavljaju ključ i vrijednost koju želimo pohraniti (a koji moraju biti *Writable*).

Sada možemo dodati nove ključeve i vrijednosti naredbom:

```
writer.append(key, value);
```

Uz pomoć metode `close()` na kraju upisa zatvaramo datoteku.

Za čitanje koristimo klasu *SequenceFile.Reader* koju stvaramo na sljedeći način:

```
SequenceFile.Reader reader = new SequenceFile.Reader(conf,
            SequenceFile.Reader.file(inputFile))
```

Vrijednosti možemo čitati uz pomoć naredbe:

```
reader.next(key, value);
```

koja će popuniti vrijednosti varijabli *key* i *value* sa pročitanim vrijednostima (važno je da tipovi podatka odgovaraju očekivanima, inače moramo koristiti metode refleksije i pretvorbe klasa). Naredba će vratiti *true* ili *false* u ovisnosti o uspješnosti čitanja a idući poziv će čitati sljedeći par elemenata.

Ovdje su dane samo osnove upravljanja datotekama tipa *SequenceFile.* Za više informacija pogledajte poveznicu:

http://hadoop.apache.org/docs/r2.6.4/api/index.html?org/apache/hadoop/io/SequenceFile.html

ZADATAK

Cilj zadatka je stvoriti novu datoteku tipa *SequenceFile*, popuniti ju podacima i onda ju uspješno pročitati.

Uspješnim rješenjem ovog zadatka steći ćete sljedeća znanja:

- **rad sa Hadoop-ovim *writable* objektima**
- **rad sa datotekama tipa *SequenceFile* koje pohranjuju objekte po principu ključ-vrijednost**

Detaljni opis zadatka:

Stvorite datoteku *ocitanja.bin* koja će simulirati niz od 100000 očitanja nekih senzora. Ključ svakog očitanja jest nasumični cijeli broj u intervalu [1,100] (simulira šifru senzora), a vrijednost je nasumični realni broj jednostruke preciznosti u intervalu [0.00, 99.99] (simulira vrijednost očitanja). Podatke pohraniti u binarnom obliku (ne kao retke teksta!).

Nakon stvaranja datoteke potrebno je izračunati srednju vrijednost očitanja svakog senzora. Uz pomoć klase *SequenceFile.Reader* prođite iterativno kroz datoteku i za svaki uređaj izračunajte aritmetičku sredinu mjerenja. Rezultate ispišite na zaslon u obliku:

```
Senzor 1:  52.453124
Senzor 2:  41.530053
...
```

NA KRAJU LABORATORIJSKE VJEŽBE OBRIŠITE DATOTEKU ocitanja.bin!